CLAIMS

What is claimed is:

1. A configuration management system comprising:

a configuration store that stores persisted information associated with an application, the persisted information comprising at least one of configuration and dependency information; and,

a configuration service component that manages access to the configuration store.

2. The system of claim 1, wherein the configuration service component receives a manifest associated with the application, the manifest comprising at least one of configuration and dependency information associated with the application, and the configuration service component stores at least some of the manifest information in the configuration store.

3. The system of claim 2, wherein the manifest is based, at least in part, upon a schema.

4. The system of claim 3, wherein the schema is XML-based.

5. The system of claim 2, wherein the manifest employing at least one of strong typing, validation, and assertions.

6. The system of claim 2, wherein the configuration service component compiles at least one of manifest information into a namespace, the configuration service component providing access to the namespace.

7. The system of claim 1, further comprising a configuration management engine that facilitates management of at least some of the configuration information.

8. The system of claim 1, the configuration service component facilitating access to a legacy store.

9. The system of claim 8, the legacy store comprising at least one of a registry and an INI configuration file.

10. The system of claim 1, the configuration service component facilitating at least one management service.

11. The system of claim 10, the management service comprising at least one of a group policy component and a roaming component.

12. The system of claim 10, the management service facilitating at least one of install, usage, servicing, uninstall, roaming, migration, setup, provisioning, policy, backup and/or restore.

13. The system of claim 1, further comprising an assertion engine that facilitates administration of a validation rule by the configuration service component.

14. The system of claim 1, further comprising a notification handler that provides information associated with a configuration change of the application to at least one of the application and another application.

15. The system of claim 1, further comprising a legacy handler that facilitates synchronization of the system with a legacy store.

16. The system of claim 1, wherein the configuration service component facilitates transacted commits for saving related changes together in the configuration store.

17. The system of claim 1, wherein the configuration service component employs at least one of ACL-based security and role-based security are provided at per-setting granularity.

18. The system of claim 1, wherein the configuration service component facilitates change logs and history.

19. An application program interface that facilitates communication with the system of claim 1.

20. The application program interface of claim 19, the application program interface facilitating at least one of discovery of an application's setting and modification of an application's setting.

21. The application program interface of claim 19, wherein the application program interface facilitates at least one of loading a namespace, saving a namespace, enumerating a namespace, consuming a setting value, enumerating a setting value, consuming metadata, enumerating metadata, querying, reverting a transaction, creating and merging an integrity constraint in the form of an assertion, registering for a change notification, and authoring a settings namespace.

22. A management service that employs an application program interface, wherein the application program interface facilitates communication with the system of claim 1.

23. The system of claim 1, wherein the configuration store comprises a joint engine technology database that stores a settings namespace.

24. The system of claim 23, wherein a namespace comprises metadata on the settings comprising types, attributes, and user context, the namespace further comprising an instance values of the settings,

25. The system of claim 24, wherein at least one of the metadata on the setting and instance values of the settings is stored for each user context.

26. The system of claim 1, wherein at least one of URI and Xpath can access a setting within a namespace as well as in between namespaces.

27. A configuration management system comprising:

a local cache that at least temporarily stores changes to configuration information associated with an application; and,

a configuration engine that facilitates communication of the changed configuration information stored in the local cache to a configuration service component.

28. A method for facilitating configuration management comprising:

receiving a manifest associated with an application, the manifest comprising at least one of configuration and dependency information;

registering the manifest; and,

storing at least some of the manifest information in a configuration store.

29. The method of claim 28, further comprising compiling a configuration section of the manifest into a namespace.

30. A computer readable medium having stored thereon computer executable instructions for carrying out the method of claim 28.

31. A method of facilitating configuration management comprising:

providing a manifest, the manifest associated with at least one of configuration and dependency information of a first application; and,

accessing a setting of an application *via* a configuration service component.

32. The method of claim 31 further comprising at least one of the following acts:

identifying settings in a namespace associated with the first application;

defining a name, a type, a description and default value for a setting;
defining other metadata for the setting;
providing a validation rule for the setting;
indicating service applicability for the setting; and,
identifying a dependency using an assertion expression.

33. The method of claim 32, further comprising at least one of the following acts:
accessing a setting associated with the first application; and,
accessing a setting associated with a second application.

34 A computer readable medium having stored thereon computer executable instructions for carrying out the method of claim 32.

35. A data packet transmitted between two or more computer components that facilitates configuration management, the data packet comprising:

a configuration manifest comprising a schema section and a metadata section, the schema section comprising an XML Schema Definitions Language (XSDL) document the skeletal structure of an application's settings' type definitions and element declarations, and, the metadata section comprising an XML instance document, validated against the XSDL schema document, which serves to decorate the settings elements with attribute tags, and to facilitate integrity constraints.

36. The data packet of claim 35, the schema section comprising at least one of a simple type, a restriction of a system's built-in scalar type, a complex type, and, a structure that can contain element content.

37. The data packet of claim 35, the configuration manifest further comprising at least one of a default, a description, a context, an access control, a validation, an exposure to manageability services and other state-related information.

38. A computer readable medium storing computer executable components of a configuration management system comprising:

a configuration service component that manages access to a configuration store, the configuration service component comprising an assertion engine component and a legacy handler component, wherein the configuration store stores persisted configuration information associated with an application.

39. A configuration management system comprising:

means for storing configuration information associated with an application; and,

means for managing access to the means for storing configuration information.